The Influence of Placement on Transmission in Distributed Computing of Boolean Functions

Ahmad Tanha, *Student Member, IEEE*, Derya Malak, *Member, IEEE*
Communication Systems Department, EURECOM, Sophia Antipolis, France
{ahmad.tanha, derya.malak}@eurecom.fr

***Abstract*—In this paper, we explore a distributed setting, where a user seeks to compute a linearly-separable Boolean function of degree M from N servers, each with a cache size M. Exploiting the fundamental concepts of sensitivity and influences of Boolean functions, we devise a novel approach to capture the interplay between dataset placement across servers and server transmissions and to determine the optimal solution for dataset placement that minimizes the communication cost. In particular, we showcase the achievability of the minimum average joint sensitivity, $\frac{N}{2^{M-1}}$, as a measure for the communication cost.**

***Index Terms*—Boolean function analysis, sensitivity, influence, distributed computing, placement-transmission tradeoffs.**

I. Introduction

Over the past few decades, technological advancements have significantly increased the demand for high-performance distributed computing to divide a computationally heavy task into multiple subtasks with lower computation load over workers across a network, e.g., machine learning algorithms over distributed servers [1], and cloud computing platforms [2]. Even though there exist heuristic approaches to the problem of distributed computing in the literature, such as MapReduce [3], managing the ever-increasing demands requires a deep understanding of distributed placement, compression, and transmission of datasets towards realizing various computations, which is our key focus in this paper.

A. Related Work

We first review functional compression literature. We then provide the existing algorithms for distributed placement of datasets to achieve various computation tasks.

a) From source compression to functional compression: While the fundamental limits for the problem of data compression, either centralized [4] or distributed [5], has been explored, the general problem of compression for computing, or functional compression, requires different tools that can exploit the structure of the computation task. To that end, Körner introduced the notion of graph entropy for distinguishing source symbols that produce different function outcomes [6], [7], and the concept of graph coloring was later used in various distributed functional compression settings, including but not limited to [8]–[10]. However, this technique may not apply to general functions. Other works, e.g., [11], [12], to exploit the characteristics of functions, devised structured coding schemes, which require different encoding functions for different tasks and hence may not be practical.

Following the coded distributed computing scheme in [13], several works investigated the storage-computation-communication tradeoffs, e.g., for the class of linearly-separable functions, using cyclic placement [14], or with linear coding for optimizing placement and transmissions [15], using placement delivery arrays [16], and tessellations [17].

b) The role of placement in distributed computation: The most common placement of datasets across servers in distributed setting's literature is the "cyclic placement" scheme on datasets, e.g., as in [14]. The placement on distributed servers is conducted in a cyclic manner, in the amount of some circular shifts between two consecutive servers. As a result of cyclic placement, any subset of servers covers the set of all datasets to compute the requested functions from the user.

While the main focus in the related literature is on encoding, transmission, and decoding phases with a given assignment of data, the placement configuration of datasets across servers could significantly affect the communication cost [15], [17].

B. Motivation and Contributions

Motivated by the impact of dataset placement on transmission in distributed computing systems [15], [17], we utilize the concept of sensitivity and influence of Boolean functions [18], [19] to designate an optimal placement configuration that achieves the minimum communication cost. Specifically, we focus on the linearly-separable Boolean functions.

In this paper, we present a novel distributed computing approach that involves a master node, a set of distributed servers, and a user demanding the error-free computation of a linearly-separable Boolean function. The master node distributes datasets across servers, where each server then performs subcomputations of datasets. Our approach captures the joint influences of subsets of distributed datasets in computing the user demand for any given number of servers with identical cache sizes. This enables us to show the fundamental interplay between the placement and transmission for distributed computation of linearly-separable Boolean functions, where the function structure reveals an optimal placement configuration.

C. Organization

The rest of the paper is organized as follows. In Section II, we present the proposed scheme for distributed computing of a

Funded by the European Union (ERC, SENSIBILITÉ, 101077361). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

This research was partially supported by a Huawei France-funded Chair towards Future Wireless Networks, and supported by the program "PEPR Networks of the Future" of France 2030.

Boolean function, and the relation between dataset placement and transmissions. Next, in Section III, exploiting the notions of sensitivity and influences of Boolean functions given placement configurations, we propose a novel approach for analyzing the communication cost for distributed computing of linearly separable Boolean functions. Finally, in Section IV, we discuss potential future directions toward extending the influence-based concept to a general class of functions.

Notation. The notation $\mathbb{F}_2^L$ represents the binary field of length L, where $\mathbb{F}_2 = \{0, 1\}$. We use square brackets to represent a set of integers, where $[K] \triangleq \{1, 2, \ldots, K\}$, given $K \in \mathbb{Z}^+$, and curly brackets to denote a set of subsets, e.g., $\{\mathcal{S}_n\}$, where $\mathcal{S}_n$ is a subset of datasets. For a random variable X, $\mathbb{E}[X]$ is its expected value. We denote by $\mathbf{W} = (W_1, \ldots, W_K)$ the vector of all datasets. The basis vector notation $e_k \triangleq (0, \ldots, 0, 1, 0, \ldots, 0)$ represents a binary vector with cardinality K such that $e_k(k) = 1$ and $e_k(l) = 0$, $\forall\ l \neq k$. The notations $\oplus$ and $\bigoplus$ indicate the modulo two addition and the summation symbol in $\mathbb{F}_2$, respectively. Hence, $\mathbf{w} \oplus e_k$ represents $\mathbf{w}$ with the k^{th} entry flipped. We denote the indicator function by $1_{\{.\}}$.

II. System Model

We consider a distributed computing setup consisting of a master node, a set of distributed servers, and a user. In this setting, there are K independent and identically distributed (i.i.d.) datasets, where each dataset $k \in [K]$ is a Bernoulli distributed random variable, denoted by $W_k \sim \text{Bern}(\frac{1}{2})$. The master node assigns (possibly not disjoint) subsets of datasets N servers indexed by $[N]$, where each server has an identical cache size that allows storing up to M datasets. Finally, the user seeks to compute a Boolean function[1] $f : \mathbb{F}_2^K \longrightarrow \mathbb{F}_2$ of the input vector of all K datasets, i.e., $\mathbf{W} = (W_1, W_2, \ldots, W_k)$.

In this paper, we use the below representation for a Boolean function in general polynomial form [19]:

$$f(\mathbf{W}) = \bigoplus_{\mathcal{P} \subseteq [K]} c_{\mathcal{P}} \prod_{k \in \mathcal{P}} W_k \tag{1}$$

for some subsets $\mathcal{P}$ of K datasets and coefficients $c_{\mathcal{P}} \in \mathbb{F}_2$.

A. Phases of Distributed Computing

In this setting, we have three phases for distributed computing of $f(\mathbf{W})$ given the input vector $\mathbf{W}$, as described next.

a) Dataset Placement: In this phase, the master node will assign a subset of datasets to each server node without coding across different datasets, known as uncoded placement in the literature, see e.g., [20]. In particular, given a cache size of M for each server, the master node will assign subsets of K datasets to the servers according to a placement function ρ_n, $n \in [N]$ which is described as

$$\rho_n : \mathbb{F}_2^K \longrightarrow \mathbb{F}_2^M \ , \quad \forall n \in [N] \ , \tag{2}$$

where the assigned subset to the n^{th} server is specified as

$$\mathcal{S}_n = \rho_n(\mathbf{W}) \subseteq \{\mathbf{W}\} \ , \quad |\mathcal{S}_n| = M \ , \quad \forall n \in [N] \ . \tag{3}$$

In other words, the placement phase assigns subsets with a cardinality M of possibly overlapping datasets to the servers.

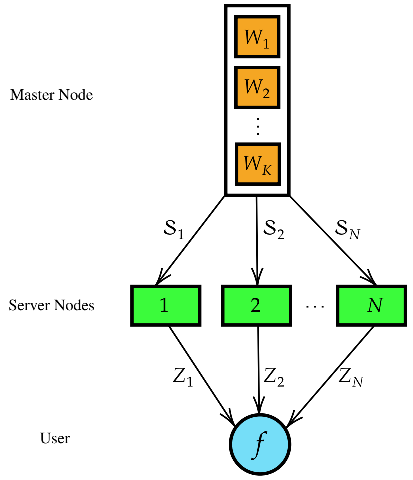

Fig. 1: A generic distributed computing system model.

We also denote the set of the assigned subsets, i.e., placement configuration by $\mathcal{S} = \{\mathcal{S}_n \,|\, n \in [N]\}$.

b) Encoding and Transmissions: Given the subsets $\mathcal{S}_n, n \in [N]$ from the placement phase, we next detail the encoding phase. Given the subset $\mathcal{S}_n$ and the computation task $f(\mathbf{W})$, server $n \in [N]$ will conduct subcomputations to determine its transmitted information. Here we note that the servers know the task $f(\mathbf{W})$ a priori and design the subcomputations accordingly. We model the encoding and transmission process at server $n \in [N]$ by the function

$$E_n^f : \mathbb{F}_2^M \longrightarrow \mathbb{F}_2^{|Z_n|} \ , \tag{4}$$

where we denote the set of computed data by server n by

$$Z_n = E_n^f(\mathcal{S}_n) = \{Z_{ni} \,|\, i \in [|Z_n|]\} \ , \quad n \in [N] \tag{5}$$

which is then transmitted to the user. We also denote by $Z = \{Z_n \,|\, n \in [N]\}$ the set of all transmitted data by all servers to measure the total number of transmissions. The user, as we describe next, will aggregate the transmissions Z from all servers to determine the output of $f(\mathbf{W})$.

c) Decoding: We assume that for any given placement configuration and given input vector $\mathbf{W}$, once the user receives the subcomputations from each server, it will be able to calculate the outcome of the Boolean function, i.e., $f(\mathbf{W})$, represented by the general polynomial form in (1). The decoding procedure should be designed based on the placement scheme and the encoding process. The decoding function for the recovery of the function by the user is specified as

$$D : \mathbb{F}_2^{|Z|} \longrightarrow \mathbb{F}_2 \ . \tag{6}$$

We assume that the user can recover the function without any error. Hence, for error-free recovery of the computation task, the decoding procedure must satisfy $D(Z) = f$.

We illustrate the system model for our distributed computing scheme in Figure 1. In this work, we focus on linearly-separable Boolean function of degree d as

$$f(\mathbf{W}) = \bigoplus_{n=1}^{N} f_n \ , \quad f_n = \prod_{k \in \mathcal{K}_{\mathcal{P}_n, d}} W_k \ , \tag{7}$$

[1] A function f is called Boolean if it only accepts binary values as both domain and range, i.e., 0 and 1 [19].

Assigned subsets	Transmitted data
$\mathcal{S}_1^{(1)}$	$Z_{11}^{(1)} = W_1W_4$, $Z_{12}^{(1)} = W_3W_6$, $Z_{13}^{(1)} = W_2$
$\mathcal{S}_2^{(1)}$	$Z_{21}^{(1)} = W_5W_8W_7$, $Z_{22}^{(1)} = W_7$, $Z_{23}^{(1)} = W_9$
$\mathcal{S}_3^{(1)}$	No transmissions

TABLE I: Server-transmission details for $\mathcal{S}^{(1)}$.

where $\mathcal{K}_{\mathcal{P},d}$ is a subset $\mathcal{P}$ with cardinality d of K datasets:

$$\mathcal{K}_{\mathcal{P},d} \triangleq \{\mathcal{P} \subseteq [K] \mid |\mathcal{P}| = d\} \ , \tag{8}$$

where $d = M$. We also assume that $K = NM$, implying N servers each with cache size M.

We refer to our system model as a $(K, N, M, \mathcal{S}, f)$ distributed computing scheme. We next define an achievable scheme for error-free distributed computing of f at the user.

Definition 1. *(***An achievable distributed computing scheme.***) A $(K, N, M, \mathcal{S}, f)$ distributed computing scheme is called achievable if the function f can be recovered in an error-free manner by the user with the given cache configuration, i.e., $D(Z) = f$, where Z is possibly a nonlinear combination of the encoded data $Z_n = E_n^f(\mathcal{S}_n)$, $n \in [N]$, which is placement-dependent and function-aware.*

Exploiting the definition of $Z = \{Z_n \,|\, n \in [N]\}$, we denote the total number of transmissions by all servers as

$$T^{(\mathcal{S})} \triangleq |Z| = \sum_{n=1}^{N} |Z_n| \ . \tag{9}$$

We next consider an example to demonstrate the interplay between the placement configuration and the value of $T^{(\mathcal{S})}$ for the given $(K, N, M, \mathcal{S}, f)$ distributed computing scheme. We will then show the connection between our model for communication cost and $T^{(\mathcal{S})}$ in Section III.

B. The Interplay between Placement and Transmission

In this subsection, we first present an example, with two different placement configurations, namely $\mathcal{S}^{(1)}$ which is cyclic, and $\mathcal{S}^{(2)}$, for computing a Boolean function. We next contrast the total number of transmissions needed in each configuration to demonstrate the role of placement and transmissions.

Example 1. *Consider a distributed computing system with 9 datasets, 3 servers, each with cache size 6, for evaluating*

$$f(\mathbf{W}) = W_1W_4W_7 \oplus W_2W_5W_8W_7 \oplus W_3W_6W_9 \tag{10}$$

with different placement configurations as follows.

(i) **Configuration** $\mathcal{S}^{(1)}$: *The dataset assignment is cyclic such that $\mathcal{S}_1^{(1)} = \{W_1, W_2, W_3, W_4, W_5, W_6\}$, $\mathcal{S}_2^{(1)} = \{W_4, W_5, W_6, W_7, W_8, W_9\}$, and $\mathcal{S}_3^{(1)} = \{W_1, W_2, W_3, W_7, W_8, W_9\}$, satisfying the cache size constraint with equality. To successfully compute (10) in this configuration, it suffices that the servers need to compute and transmit data as shown in Table I. Hence, this scenario requires $T^{\mathcal{S}^{(1)}} = 6$ transmissions in total.*

(ii) **Configuration** $\mathcal{S}^{(2)}$: *The dataset assignment satisfies $\mathcal{S}_1^{(2)} = \{W_1, W_2, W_4, W_5, W_6, W_7\}$, $\mathcal{S}_2^{(2)} = \{W_3, W_4, W_5, W_6, W_8, W_9\}$, and $\mathcal{S}_3^{(2)} = \{W_1, W_2, W_3, W_7, W_8, W_9\}$. To successfully compute (10) in this setting, without the presence of stragglers, a viable transmission scheme is shown in Table II. Thanks to a better arrangement of the datasets that is sensitive to the function in $\mathcal{S}^{(2)}$ versus $\mathcal{S}^{(1)}$, the total number of transmissions is $T^{(\mathcal{S}^{(2)})} = 4 < T^{(\mathcal{S}^{(1)})} = 6$.*

Assigned subsets	Transmitted data
$\mathcal{S}_1^{(2)}$	$Z_{11}^{(2)} = W_1W_4W_7$, $Z_{12}^{(2)} = W_2W_5W_7$
$\mathcal{S}_2^{(2)}$	$Z_{21}^{(2)} = W_3W_6W_9$, $Z_{22}^{(2)} = W_8$
$\mathcal{S}_3^{(2)}$	No transmissions

TABLE II: Server-transmission details for $\mathcal{S}^{(2)}$.

We infer from Example 1 how a cleverly conducted placement phase in distributed computing settings could dramatically reduce the total number of transmissions needed for error-free recovery of the Boolean function at the user.

To evaluate the communication cost of the $(K, N, M, \mathcal{S}, f)$ distributed computing scheme, we will next detail a novel approach that relies on the average joint sensitivity of the computation task abstracted by the Boolean function.

III. Main Results

In this section, to determine the role of the placement configuration on the communication cost, we first provide a primer on the sensitivity of a Boolean function on its input, and the influence of a set of input variables on the outcome of the function. We then present our main results.

A. Joint Sensitivity and Influences

Building on the classical notions of sensitivity, influence, and average sensitivity tailored for capturing the sensitivity of a Boolean function by modifying one dataset at each time [18], [19], we will exploit the joint behavior of datasets across servers, as in [21]. A Boolean function $f(\mathbf{W})$ depends on its k^{th} input variable if there exists at least one $\mathbf{W} \in \mathbb{F}_2^K$ such that $f(\mathbf{W} \oplus e_k) \neq f(\mathbf{W})$. To that end, we next define the sensitivity of $f(\mathbf{W})$ on a set of W_k's.

Definition 2. (Joint sensitivity.) *The joint sensitivity of $f(\mathbf{W})$ to the set $\mathcal{S}$ of subsets of datasets is defined as*

$$\mathrm{Sen}_{\mathcal{S}}(f, \mathbf{W}) = \sum_{n=1}^{|\mathcal{S}|} \mathrm{Sen}_{\mathcal{S}_n}(f, \mathbf{W}) \ , \tag{11}$$

where $e_{\mathcal{S}_n} \triangleq \bigoplus_{\{k | w_k \in \mathcal{S}_n\}} e_k$ describes the multi-dataset flipped vector, the measure $\mathrm{Sen}_{\mathcal{S}_n}(f, \mathbf{W}) = 1_{f(\mathbf{W} \oplus e_{\mathcal{S}_n}) \neq f(\mathbf{W})}$ captures the sensitivity of $f(\mathbf{W})$ on input $\mathbf{W}$ with the jointly flipped entry datasets with indices k such that $W_k \in \mathcal{S}_n \subseteq \mathcal{S}$.

Definition 3. (Joint influence.) *The joint influence of the datasets of $\mathcal{S}_n$ on the function f is defined as*

$$\mathrm{Inf}_{\mathcal{S}_n}(f) = \mathbb{P}[f(\mathbf{W} \oplus e_{\mathcal{S}_n}) \neq f(\mathbf{W})] = \mathbb{E}_{\mathbf{W}}[\mathrm{Sen}_{\mathcal{S}_n}(f, \mathbf{W})] \ .$$

Definitions 2-3 allow us to introduce our next key metric.

Definition 4. (Average joint sensitivity.) *The average joint sensitivity of $f(\mathbf{W})$ to the set $\mathcal{S} = \{\mathcal{S}_n | n \in [N]\}$ of all possible* *datasets specified by ρ_n given a cache size constraint M is given as follows:*

$$\mathrm{as}_{\mathcal{S}}(f) = \mathbb{E}_{\mathbf{W}}[\mathrm{Sen}_{\mathcal{S}}(f, \mathbf{W})] = \sum_{n=1}^{|\mathcal{S}|} \mathrm{Inf}_{\mathcal{S}_n}(f) \ . \tag{12}$$

Utilizing Definitions 2-4, we can next present our approach for formulating an optimal dataset placement configuration from a communication cost perspective.

B. The Communication-Optimal Placement Configuration

To evaluate the tradeoff between placement and communication cost for computing a Boolean function f, we first present a Lemma that focuses on each product subfunction f_n given in (7) to obtain the joint influence of datasets on f_n.

Lemma 1. (Joint influence on a product subfunction.) *The joint influence of multiple datasets in a subset $\mathcal{S}_n$ with an arbitrary size from a product subfunction $f_n(\mathbf{W}) = \prod_{k \in \mathcal{K}_{\mathcal{P},d}} W_k$ of degree d equals the influence of each dataset on f_n, i.e.,*

$$\mathrm{Inf}_{\mathcal{S}_n}(f) = \mathrm{Inf}_k(f) = \frac{1}{2^{d-1}} \ . \tag{13}$$

Proof. See Appendix A. □

Towards determining the joint influence of datasets on f in (7), we next present another Lemma that contrasts the joint influence of datasets for the summation of two product subfunctions, namely f_n and $f_{n'}$ where $n' \neq n$, for different dataset placement configurations. To that end, we denote the set of variables included in the subfunction f_n as

$$\mathcal{I}_{f_n} \triangleq \{W_k \,|\, f_n = \prod_{k \in \mathcal{K}_{\mathcal{P}_n,d}} W_k\}, \quad \forall n \in [N] \ . \tag{14}$$

Lemma 2. (Increase in joint influence due to summation.) *Let $f = f_n \oplus f_{n'}$, where $n \neq n'$. Consider two different subsets of datasets denoted by $\mathcal{S}_1 = \{W_k | W_k \in \mathcal{I}_{f_n}\}$ and $\mathcal{S}'_1 = \{W_k | W_k \in \mathcal{I}_{f_n} \cup \mathcal{I}_{f_{n'}}\}$. We then have:*

$$\mathrm{Inf}_{\mathcal{S}'_1}(f) \geq \mathrm{Inf}_{\mathcal{S}_1}(f) \ . \tag{15}$$

Proof. See Appendix B. □

From Lemma 2, the subsets with the lowest joint influence include datasets from the same f_n. We next derive a lower bound on the average joint sensitivity for the proposed setting using Lemmas 1-2, and present the communication-optimal placement configuration in Theorem 1.

Theorem 1. (A communication-optimal placement configuration.) *Given a $(K, N, M, \mathcal{S}, f)$ distributed computing scheme, the average joint sensitivity and the total number of transmissions are lower bounded by*

$$\mathrm{as}_{\mathcal{S}}(f) \geq \mathrm{as}_{\mathcal{S}^*}(f) = \frac{N}{2^{M-1}} \ , \quad T^{(\mathcal{S})} \geq T^{(\mathcal{S}^*)} = N \ , \tag{16}$$

respectively, corresponding to $\mathcal{S}^* = \{\mathcal{S}^*_n \,|\, n \in [N]\}$, where $\mathcal{S}^*_n = \{W_k \,|\, W_k \in \mathcal{I}_{f_n}\}$.

Proof. See Appendix C. □

Corollary 1. *The number of transmissions by server n is a monotonically increasing function in terms of the joint influence of datasets in each subset on Boolean function f, i.e., $|Z_n| \leq |Z_{n'}|$ if and only if $\mathrm{Inf}_{\mathcal{S}_n}(f) \leq \mathrm{Inf}_{\mathcal{S}_{n'}}(f), \forall n \neq n'$.*

From Corollary 1, it is easy to observe that $T^{(\mathcal{S})}$ is a monotonically increasing function of the average joint sensitivity, i.e., $T^{(\mathcal{S})} \leq T^{(\mathcal{S}')}$ if and only if $\mathrm{as}_{\mathcal{S}}(f) \leq \mathrm{as}_{\mathcal{S}'}(f)$.

IV. Conclusions

In this work, using the concept of sensitivity and influence, we introduced a novel approach for determining the interplay between communication cost and placement configuration for distributed computing of Boolean functions. In particular, we specified the optimal placement configuration from a communication cost perspective for a class of linearly-separable Boolean functions. Our approach is based on grouping the datasets to minimize the summation of their joint influences. As a future direction, we will extend our approach to nonlinear Boolean functions of any degree.

Appendix

A. Proof of Lemma 1

Using Definition 3 and its probabilistic nature, we have:

$$\mathrm{Inf}_{\mathcal{S}_n}(f) = \mathbb{P}\Big[f(\mathbf{W} \oplus e_{\mathcal{S}_n}) \neq f(\mathbf{W})\Big] \overset{(*)}{=} \frac{2}{2^d} = \frac{1}{2^{d-1}} \ ,$$

where $(*)$ holds since only two sequences $11\ldots1$ and $00\ldots0$ out of 2^d possible sequences are feasible. Similarly, for the individual variable W_k, the influence is calculated as

$$\mathrm{Inf}_k(f) = \mathbb{P}\Big[\prod_{i \in \mathcal{K}_{\mathcal{P} \backslash k, d-1}} W_i \neq 0\Big] = \Big(\frac{1}{2}\Big)^{d-1} \ , \tag{17}$$

where the last step follows when all the $d-1$ datasets $W_k \sim \mathrm{Bern}(\frac{1}{2})$, $k \in \mathcal{K}_{\mathcal{P} \backslash k, d-1}$ are equal to 1. Therefore, (13) holds.

B. Proof of Lemma 2

According to Lemma 1, $\mathrm{Inf}_{\mathcal{S}_1}(f) = \frac{1}{2^{d-1}}$. To find the joint influence of subset $\mathcal{S}'_1$ on $f = f_n \oplus f_{n'}$, we first consider only one dataset difference between $\mathcal{S}_1$ and $\mathcal{S}'_1$, i.e., we assume $\mathcal{S}'_1 = \{W_k | W_k \in \mathcal{I}_{f_n} \backslash W_n \cup W_{n'}\}$. We then decompose the respective product subfunctions as

$$f_j = \prod_{k \in \mathcal{K}_{\mathcal{P}_j \backslash j, d-1}} W_k W_j \ , \quad j \in \{n, n'\} \ .$$

Using Definition 3, we can rewrite $\mathrm{Inf}_{\mathcal{S}'_1}(f)$ as

$$\begin{aligned} \mathrm{Inf}_{\mathcal{S}'_1}(f) = \mathbb{P}[& \prod_{k \in \mathcal{K}_{\mathcal{P}_n \backslash n, d-1}} (W_k \oplus 1) W_n \\ & \oplus \prod_{k \in \mathcal{K}_{\mathcal{P}_{n'} \backslash n', d-1}} W_k (W_{n'} \oplus 1) \\ & \neq \prod_{k \in \mathcal{K}_{\mathcal{P}_n \backslash n, d-1}} W_k W_n \oplus \prod_{k \in \mathcal{K}_{\mathcal{P}_{n'} \backslash n', d-1}} W_k W_{n'}] \\ & = \mathbb{P}\Big[\bigoplus_{i=0}^{d-2} \prod_{k \in \mathcal{K}_{\mathcal{P}_n \backslash n, i}} W_k W_n \oplus \prod_{k \in \mathcal{K}_{\mathcal{P}_{n'} \backslash n', d-1}} W_k \neq 0\Big] \ . \end{aligned} \tag{18}$$

Considering $\Delta f \triangleq \bigoplus_{i=0}^{d-2} \prod\limits_{k\in\mathcal{K}_{\mathcal{P}_n\setminus n,i}} W_k W_n \oplus \prod\limits_{k\in\mathcal{K}_{\mathcal{P}_{n'}\setminus n',d-1}} W_k$, the law of total probability, and i.i.d. datasets:

$$\mathbb{P}[\Delta f=1] = \frac{1}{2}\left(\mathbb{P}[\Delta f=1\,|\,W_n=1]\right. \\ \left.+\mathbb{P}[\Delta f=1\,|\,W_n=0]\right) ,$$

where using Lemma 1, we obtain:

$$\mathbb{P}[\Delta f=1\,|\,W_n=0] = \mathbb{P}\Big[\prod_{k\in\mathcal{K}_{\mathcal{P}_{n'}\setminus n',d-1}} W_k=1\Big] = \left(\frac{1}{2}\right)^{d-1} ,$$

$$\begin{aligned}
&\mathbb{P}[\Delta f=1\,|\,W_n=1] \\
&=\mathbb{P}\Big[\bigoplus_{i=0}^{d-2}\prod_{k\in\mathcal{K}_{\mathcal{P}_n\setminus n,i}} W_k \oplus \prod_{k\in\mathcal{K}_{\mathcal{P}_{n'}\setminus n',d-1}} W_k=1\Big] \\
&=\mathbb{P}\Big[\bigoplus_{i=0}^{d-2}\prod_{k\in\mathcal{K}_{\mathcal{P}_n\setminus n,i}} W_k=0, \prod_{k\in\mathcal{K}_{\mathcal{P}_{n'}\setminus n',d-1}} W_k=1\Big] \\
&+\mathbb{P}\Big[\bigoplus_{i=0}^{d-2}\prod_{k\in\mathcal{K}_{\mathcal{P}_n\setminus n,i}} W_k=1, \prod_{k\in\mathcal{K}_{\mathcal{P}_{n'}\setminus n',d-1}} W_k=0\Big] \\
&=\Big[1-\left(\frac{1}{2}\right)^{d-2}\Big]\left(\frac{1}{2}\right)^{d-1}+\left(\frac{1}{2}\right)^{d-2}\Big[1-\left(\frac{1}{2}\right)^{d-1}\Big] . \qquad (19)
\end{aligned}$$

After rearranging (19) and applying it to (18), we obtain:

$$\mathrm{Inf}_{\mathcal{S}_1'}(f) = 2\left(\frac{1}{2}\right)^{d-1}\Big[1-\left(\frac{1}{2}\right)^{d-1}\Big] . \qquad (20)$$

It is obvious that $\mathrm{Inf}_{\mathcal{S}_1'}(f) \geq (\frac{1}{2})^{d-1} = \mathrm{Inf}_{\mathcal{S}_1}(f)$. By induction, it follows that for any subset $\mathcal{S}_1'$ with more difference than one dataset compared to $\mathcal{S}_1$, (15) holds.

C. Proof of Theorem 1

We prove this theorem in two parts:

a) Achievability: For distributed computing of a linearly-separable Boolean function f of degree M, there exists an achievable scheme with a placement configuration $\mathcal{S}^*$.

According to Lemma 1, $\mathrm{Inf}_{\mathcal{S}_n^*}(f) = \frac{1}{2^{M-1}}, \forall n \in [N]$. The average joint sensitivity in this case is therefore:

$$\mathrm{as}_{\mathcal{S}^*}(f) = \sum_{n=1}^{N}\frac{1}{2^{M-1}} = \frac{N}{2^{M-1}} . \qquad (21)$$

According to Definition 1, the $(K,N,M,\mathcal{S}^*,f)$ distributed computing scheme is achievable since the user with cache configuration $\mathcal{S}^*$ would be able to recover the function in an error-free manner with only summing Z_n^*'s together, i.e.,

$$D(Z^*) = \bigoplus_{n=1}^{N} Z_n^* = \bigoplus_{n=1}^{N}\prod_{k\in\mathcal{K}_{\mathcal{P}_n^*,M}} W_k = \bigoplus_{n=1}^{N} f_n = f . \qquad (22)$$

b) Optimality (converse): Based on Lemma 2, the minimum joint influence happens when we group the datasets from the same product subfunction. We then use it to show the optimality of our proposed placement configuration.

We then examine a placement scheme $\mathcal{S}$, where we consider similar placement as $\mathcal{S}^*$ for $N-2$ servers. For simplicity, we only swap two datasets between the two first servers. They will therefore contain subsets $\mathcal{S}_1 = \{W_k\,|\,W_k \in \mathcal{I}_{f_n}\setminus W_n \cup W_{n'}\}$ and $\mathcal{S}_2 = \{W_k\,|\,W_k \in \mathcal{I}_{f_{n'}}\setminus W_{n'} \cup W_n\}$, respectively, i.e., we group $M-1$ datasets from $\mathcal{I}_{f_n}$ and one variable ($W_{n'}$) from $\mathcal{I}_{f_{n'}}$ in $\mathcal{S}_1$ and vice versa in $\mathcal{S}_2$. According to Lemma 1 and Lemma 2, we have:

$$\mathrm{Inf}_{\mathcal{S}_n}(f) \geq \mathrm{Inf}_{\mathcal{S}_n^*}(f) = \frac{1}{2^{M-1}},\ \forall n=1,2 . \qquad (23)$$

For other $N-2$ subsets, we also have:

$$\mathrm{Inf}_{\mathcal{S}_n}(f) = \mathrm{Inf}_{\mathcal{S}_n^*}(f) = \frac{1}{2^{M-1}} . \qquad (24)$$

Using Definition 4 and summing the joint influences together for both $\mathcal{S}$ and $\mathcal{S}^*$ completes the proof. By induction, it follows that for any placement configuration $\mathcal{S}$ with more swapped datasets between subsets compared to $\mathcal{S}^*$, (16) holds. We also note that the minimum value of $T^{(\mathcal{S})}$ corresponds to $\mathcal{S}^*$ and equals to N, i.e., the transmission of N units of data.

References

[1] H. Li, Y. Chen, K. W. Shum, and C. W. Sung, "Data allocation for approximate gradient coding in edge networks," in *Proc., IEEE ISIT*, Taipei, Taiwan, 2023, pp. 2541–2546.

[2] X. Xu, M. Tang, and Y.-C. Tian, "Theoretical results of QoS-guaranteed resource scaling for cloud-based MapReduce," *IEEE Trans. Cloud Comput.*, vol. 6, no. 3, pp. 879–889, 2018.

[3] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, p. 107–113, Jan. 2008.

[4] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[5] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[6] J. Körner, "Coding of an information source having ambiguous alphabet and the entropy of graphs," in *Proc., 6th Prague Conf. Inf. Theory, Prague, Czech Republic*, Sep. 1973, pp. 411–425.

[7] M. Sefidgaran and A. Tchamkerten, "Computing a function of correlated sources: A rate region," in *Proc., IEEE ISIT*, St. Petersburg, Russia, 2011, pp. 1856–1860.

[8] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–17, Mar. 2001.

[9] S. Feizi and M. Médard, "On network functional compression," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5387–5401, Jun. 2014.

[10] D. Malak, "Distributed computing of functions of structured sources with helper side information," in *Proc., IEEE SPAWC*, Shanghai, China, Sep. 2023, pp. 276–280.

[11] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.

[12] T. Han and K. Kobayashi, "A dichotomy of functions F (X, Y) of correlated sources (X, Y)," *IEEE Trans. Inf. Theory*, vol. 33, no. 1, pp. 69–76, Jan. 1987.

[13] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, 2018.

[14] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1259–1278, Feb. 2022.

[15] A. Khalesi and P. Elia, "Multi-user linearly-separable distributed computing," *IEEE Trans. Inf. Theory*, vol. 69, no. 10, pp. 6314–6339, 2023.

[16] Q. Yan, S. Yang, and M. Wigger, "Storage-computation-communication tradeoff in distributed computing: Fundamental limits and complexity," *IEEE Trans. Inf. Theory*, vol. 68, no. 8, pp. 5496–5512, 2022.

[17] A. Khalesi and P. Elia, "Tessellated distributed computing," 2024. [Online]. Available: https://arxiv.org/abs/2404.14203

[18] J. Bourgain, J. Kahn, G. Kalai, and et. al., "The influence of variables in product spaces," *Isr. J. Math.*, vol. 77, pp. 55–64, 1992.

[19] S. Jukna, *Boolean function complexity: advances and frontiers*. Springer, 2012, vol. 5.

[20] S. Jin, Y. Cui, H. Liu, and G. Caire, "Uncoded placement optimization for coded delivery," in *Proc., IEEE WiOpt*, Shanghai, China, May 2018.

[21] A. Biswas and P. Sarkar, "Influence of a set of variables on a Boolean function," *SIAM J. Discrete Math.*, vol. 37, no. 3, pp. 2148–2171, 2023.